Practice Management System

Integration Guide

Dental Desktop version 1.7.0.0 and higher

3shape

Contents

Introduction

TRIOS is available on the Dental Desktop (*3DD*) platform, which has the option to integrate with Practice Management Systems (*PMS*). This integration allows for a seamless flow of patient and case information between the PMS partner software and Dental Desktop. The integration comes in 3 interfaces: Command line (*CLI*), VDDS and REST API. This document consists of detailed information on each integration type.

An integration of this kind allows users to manage and create new patients in their PMS software and later, sync this information with the Dental Desktop when they wish to scan the patients. When the patient has been scanned it is possible to sync images and case information back to the PMS, to achieve a full overview directly in the software.

The integration technology is made completely open, which means that any PMS provider can make an integration to the Dental Desktop software platform. In this document, you will find the information about what steps are required for PMS providers to start their very own integration.

How to integrate

If the PMS provider has not already done so, they should reach out to PMS@3shape.com and provide the following:

- Email subject: PMS Integration with 3Shape
- Company details: name, address and CVR/VAT number
- Contact person: name, contact information, role in the company
- Type of integration: state which type of integration you wish to create (*REST, CLI or VDDS*)
- 3Shape dongle: dongle number, if you have one. If no dongle, request a dongle with a 3Shape software trial license.
- Key markets or market reach: please specify either region (*Global, EU, Americas, Asia Pacific, Scandinavia, etc.*) or countries (*US, France, Australia, China, etc.*).
- Forward any initial questions you may have based on the information you've seen so far.
- 3DD: Request a link to download the software.

Once 3Shape receives this email, a technical consultant will get in touch with the contact person (*and sender of the email*) to confirm the next steps to get the 3Shape software installed, and up and running for the PMS provider.

Once the integration is established and released, please notify 3Shape. 3Shape will then include the integration in a detailed overview online, at trade shows, etc., with information about your company and with a link to any specific instructions and/or 'how-to guide' required to make the integration work for end-users.

Important note:

- Information that is potentially sensitive (*e.g. patient name or patient ID etc.*) leaves the secure Dental Desktop environment, when using the Practice Management Integration module, to integrate with external systems.

Choosing the right integration interface

3Shape supports 3 different integration types and choosing the right one for your PMS and clients is important. This section briefly covers the differences and limitations of each interface, so the choice becomes clearer.

REST

The REST API is the most advanced integration interface and supports all current integration functionalities, such as managing patients, starting and retrieving cases, getting images, and so on. This interface does require a bit of development, but as it is considered a simple REST API, it should only be a small to medium project depending on API maturity and capacity of the PMS supplier. The API follows best practice for such interfaces.

As this interface integrates via a webservice, it will support all clinic configurations, i.e. having PMS on one computer and 3DD on another, or a cloud-hosted PMS. Furthermore, if the clinic is using TRIOS MOVE+, this is also supported.

3Shape recommends all PMS suppliers to use this interface, as it will provide the best experience for the end-users and gives them the most value. Also, the general trend is to integrate using a REST API.

Read more about 3Shape's Webservice (REST) integration.

VDDS

VDDS is primarily a German industry standard for communicating between different dental systems. It supports managing patients and retrieving case information and images. If the PMS already supports VDDS, it will be a small investment to configure/develop a connection to 3DD.

As this integration makes use of a simpler interface, not all clinic configurations are supported. It currently requires both PMS and 3DD to be installed on the same computer, so cloud-configurations are not supported. Nor can a clinic currently use TRIOS MOVE+ with a direct connection to the PMS.

3Shape recommends to only use this interface if the PMS already supports VDDS and if the clinics already use the same computer for PMS and 3DD. If the PMS supplier has to develop support for VDDS, then 3Shape recommends developing a REST webservice instead.

Read more about 3Shape's VDDS integration.

CLI

The command-line (*CLI*) interface is the simplest way to integrate the PMS to 3DD. It currently only supports patient management, such as creating and updating patient information in 3DD. There is no support for data/images from 3DD to the PMS. If the PMS already supports command-line connections, then it requires little to no development to support CLI.

As this integration interface makes use of a command-prompt, it comes with all the accompanying limitations. So not all clinic configurations are supported. It currently requires both PMS and 3DD to be installed on the same computer, so cloud-configurations are not supported. Nor can a clinic use MOVE+ with a direct connection to the PMS.

3Shape recommends to only use this interface if there is no need for 3DD images/case information, if the PMS already supports CLI, and if the clinics already use the same computer for PMS and 3DD. If the PMS supplier has to develop support for a command-prompt integration, then 3Shape recommends developing a webservice instead.

Read more about 3Shape's Command Line Integration.

Webservice (REST) integration

Dental Desktop (*3DD*) provides limited possibility to manage patients and cases, as well as obtaining cases information, by creating a webservice using Dental Desktop's REST API. The REST API is part of the 3DD server and is included with all 3Shape TRIOS installations.

Terminology

Some of the terms used in the following text is explained:

- **Client**: 3DD is (usually) installed as a client/server configuration, which means a clinic will have 1 server and *n* clients connected to this server. Each client is a computer with 3DD installed. If a clinic has standalone Trios computers, such as MOVE+, that station is both a server and a client.
- **Request**: Each use of the integration method described in this section of the guide will create a request on the 3DD server. The request only exists until it's handled by the server.
- **Case**: A case in the 3DD context can represent a new treatment plan or a new scan of a patient.

Concept of patient mapping

When Practice Management System (*PMS*) performs an operation related to a PMS patient and using the integration API (*e.g. updating 3DD patient according to new PMS patient information, or creating a new case for 3DD patient*), it is important for 3DD to know which 3DD patient corresponds to the PMS patient entry.

For this purpose, PMS patients must be mapped to 3DD patients using the **integration ID**. When performing a patient-related operation, PMS must provide the integration ID related to the corresponding PMS patient. Any PMS patient entry value can be used as the integration ID, if it's unique for each patient.

It is possible to assign an integration ID to a 3DD patient using the `CreatePatient` method, either by creating a new patient or by merging an existing one (*read /CreatePatient section for details*). **Each 3DD patient must have a unique integration ID**.

Supported workflows in REST API

- Getting the list of 3DD clients connected to 3DD server;
- Creating new 3DD patients mapped to the PMS patients;
- Mapping existing 3DD patients to the PMS patients;
- Updating mapped 3DD patients according to the changes to corresponding PMS patients;
- Selecting mapped 3DD patient on the 3DD client;
- Starting new case workflows for mapped 3DD patients on 3DD client;
- Selecting existing case on 3DD client;
- Getting information about cases;
- Fetching clinical photos from the 3DD cases;

Client/server operations

Some operations are executed on server, some can be executed only on client, while some can be executed either on client or server. The exact effect of an operation may vary depending on where it is performed, this behavior will become clear from the details in this guide. For each operation that is to be performed on a 3DD client, requests are created in a request pool on the server and is removed from the pool once the client picks it up to execute. Usually requests are executed immediately, but there might be minor delay in some clinic configurations (*usually less than second*).

It's possible to check the request execution status and remove unhandled requests if needed, for instance, *if a request was picked up by the client or not; please refer to the "**Request Status Reference**" section for details*).

Integration implementation

Performing calls to integration API

The integration methods are called via HTTP(S) requests using any convenient technology or framework (e.g. **`HttpClient`** on .NET Framework). It is also possible to call methods from any web browser. The URL for method execution should be composed as follows:

```
https://<pc-address>:5484/DentalDesktop/WebService/<method>
```

Where `<pc-address>` is a name or IP address (*preferably full PC domain name*) of the PC that hosts the 3DD server, and `<method>` is the method signature including specified parameters. If parameters are optional, they can be omitted from the URL (*in this case, default value will be used, e.g. empty string or GUID*). The parameter names are not case sensitive, *but the parameter values are*. The date and time parameters must be specified in an invariant format. The order of the parameters in the URL is not important.

Operations feedback

The operations performed on the server are mostly done instantly. However, the operations that are performed on the 3DD client might take some time to get executed, which is why it is impossible to track results of those operations instantly. These operations provide the possibility to assign a request ID to them, which can be used to poll the current operation status, *including the reason for the failure, if any*.

Security

Authorization header

Integration API uses the Basic authorization approach. Any unauthorized request will result in a 401 error returned to caller. To receive an authorization, requests must have the `Authorization` header value configured by Basic standard:

```
Basic <encrypted credentials>
```

Where `<encrypted credentials>` are credentials of any 3DD user, or a 3Shape Communicate account associated with any 3DD user, formatted as `<username>:<password>` and encoded as Base64 string.

For example, Authorization header value for the user with username "Sam" and password "qwerty123" will be:

```
Basic U2FtOnF3ZXJ0eTEyMw==
```

Certificate validation

All calls to the integration API are encrypted as HTTPS using a certificate. The certificate can be validated by comparing the thumbprint of a certificate authority (last certificate in trust chain):

737842e3e3d642d39f93d8c2d84b2ef809cd571d

Example C# code for validating the certificate from the server:

```
System.Net.ServicePointManager.ServerCertificateValidationCallback +=
(se, cert, chain, sslerror) =>
{
    var certificateAuthority = chain.ChainElements[chain.ChainElements.Count - 1].Certificate;
    return certificateAuthority != null &&
           certificateAuthority.Thumbprint == "737842e3e3d642d39f93d8c2d84b2ef809cd571d";
};
```

Security protocol

The minimal supported security protocol for contacting PMS API is TLS 1.0.

Integration API reference

In the following text, you will find all the supported methods via REST API, including a description of the method action, its input parameters and output data.

/GetAvailableClientList

Summary

This method obtains information about the 3DD clients, that are connected to the server.

Input

A full URL to call the method with the example parameters:

https://localhost:5484/DentalDesktop/WebService/GetAvailableClientList

Output

The XML string that can be deserialized for collecting the **ClientInfo** entries, corresponding to the 3DD clients connected to the server.

An example of the output received:

```
▼<ArrayOfClientInfo xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  ▼<ClientInfo>
    ▼<OperatorId>
       c3015f21-29d2-4ba2-9901-009f9859528f_364b9e1d-ec02-4b74-ab4b-20a250855d82
     </OperatorId>
     <TimeStamp>2019-11-14T13:14:20.7395144+02:00</TimeStamp>
     <ClientId>5be65d15-31a2-442b-a93b-0bd65ec8abe9</ClientId>
     <Hostname>UA-DTC-VHR</Hostname>
     <IsBusy>true</IsBusy>
    ▼<Modules>
      ▼<ModuleInfo>
         <Id>ThreeShape.TRIOS</Id>
         <Title>TRIOS</Title>
       </ModuleInfo>
     </Modules>
     <OperatorName>Default operator</OperatorName>
  </ClientInfo>
</ArrayOfClientInfo>
```

ClientInfo class contains the following fields:

- **ClientId** (GUID) - ID of the client connected to the server; can be used to determine what client will process the operation;
- **Hostname** (String) – The client's PC hostname;
- **IsBusy** (Boolean) – indicates if the client is occupied by a user or not. This does not prevent the client from executing requests. The client is considered occupied in any of the following conditions:
    - The client is currently in the case workflow;
    - The client's PC was used less than some time ago (*user's input was detected*). The actual time can be specified in the settings (*see fig.1*);
    - Background case operation is in progress (*e.g. post-processing*);
    - If time is set to 0, the user's input will be ignored;

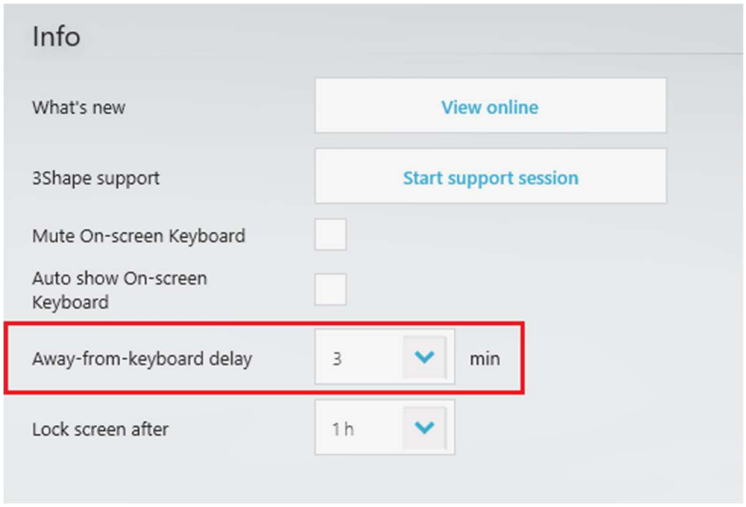

Figure 1. Setting for time of inactivity to consider client as non-busy

- **Modules** – a collection of ModuleInfo entries, corresponding to the modules installed on the client (*see ModuleInfo reference below*);
- **OperatorName** (String) – the name of the user that is logged in on the client;
- **OperatorId** (String) – ID of the user that is logged in on the client;
- **TimeStamp** (DateTime) – a timestamp of when the client information was generated;

ModuleInfo class contains following fields:

- **Id** (String) – a persistent ID of the module, could be used for module recognition;
- **Title** (String) – the full name of the module;

Remarks

The primary intended usage of this method is to provide a list of the available 3DD clients to the PMS, so the PMS user can decide which 3DD client will execute the operation.

/CreatePatient

Summary

This method allows to create a new patient (or map the PMS patient to an existing 3DD patient) either on the 3DD client or on the 3DD server.

Input

The full URL to call the method with example parameters:

```
https://localhost:5484/DentalDesktop/WebService/CreatePatient?
FirstName=John&
LastName=Smith&
DateOfBirth=14.11.2019&
PatientId=123&
IntegrationId=123&
ClientId=5be65d15-31a2-442b-a93b-0bd65ec8abe9&
RequestId=2446cb63-2f79-4f4d-9da6-c24a17755bc5&
MergePatient=False
```

This method can accept the following parameters:

- **FirstName** (**String**) – the first name of the patient to be created;
- **LastName*** (**String**) – the last name of the patient to be created. A minimal requirement for the patient creation in 3DD;
- **DateOfBirth** (**DateTime**) – the date of birth of the patient to be created;
- **PatientId** (**String**) – can be used to specify any visual patient-specific ID (*e.g. SSN*). If specified, **this value must be unique;**
- **IntegrationId*** (**String**) – internal ID of the patient. It should be used for mapping between PMS patient entry and the 3DD patient. **This value must be unique;**
- **ClientId** (**GUID**) – The ID of the 3DD client where the patient creation will occur. If the ID is not specified, *or set to default empty GUID*, the operation will be performed on the server;
- **RequestId** (**GUID**) – the optional ID for the operation that can be used to identify the operation status;
- **MergePatient** (**Boolean**) - relevant only for the client operations. If it's set to true, the 3DD client will show a merging dialog, allowing the integration ID to be assigned to an already existing 3DD patient instead of creating a new one. If not specified, the default value is **False**;

The parameters marked by ***** asterisk are mandatory and the other parameters can be omitted, in which case the *default value will be used*.

Output

This method does not provide any output.

Remarks

The parameters of this method must satisfy following the conditions:

- **LastName** parameter must be specified and not have an empty value;
- **IntegrationId** parameter must be specified, not have an empty value and the value must be unique;

- If the **`PatientId`** parameter is specified, the value must be unique.

If any of those conditions are not satisfied, the call to the method will fail, creating a corresponding request status (*see `GetRequestStatusById` method for details*).

If **`ClientId`** is not specified, new patient creation operation will be executed on the 3DD server. If operation is executed on client, it will have different effect depending on the value in **`MergePatient`** parameter.

If the **`MergePatient`** parameter is not specified or if it's set to **`False`**, the 3DD client will open a default "New Patient" dialog where the user will see all the patient data specified in the request (*see fig.2*). By selecting "Save patient", a new patient will be created, and the **integrationID** will be assigned to the created patient.

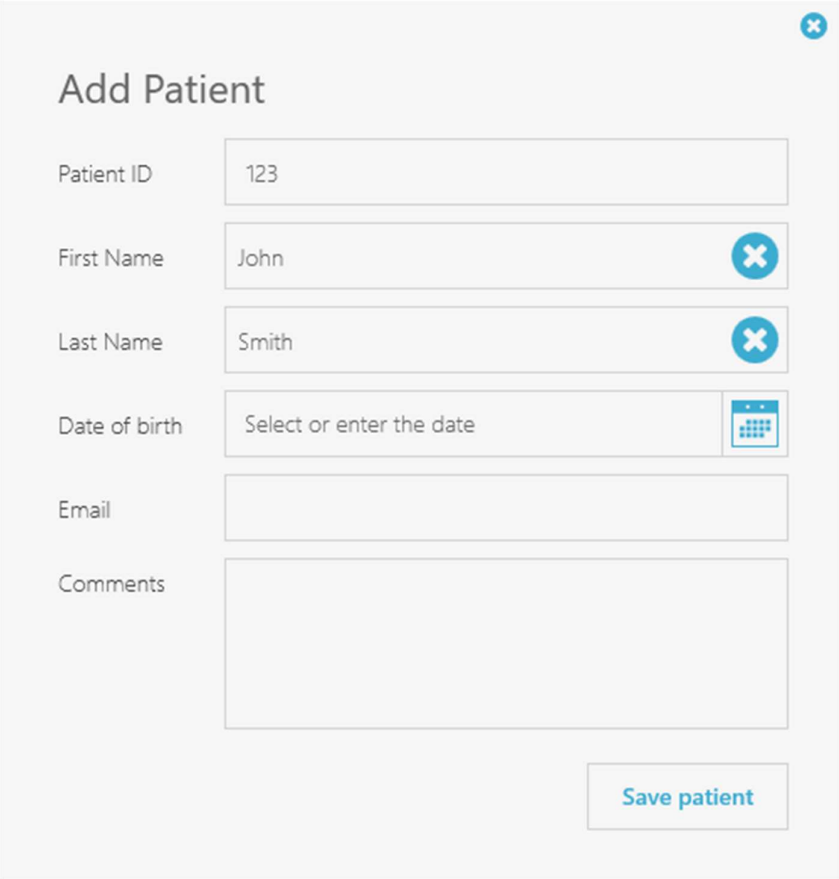

Figure 2. Patient creation dialog

If **`MergePatient`** is set to **`True`**, the 3DD client will perform a search of patients with similar first and last names to ones specified in the request (*patients, that already have **integrationID** assigned to them, will be excluded*). If any patients are found, the 3DD client will show the user the merging dialog (*see fig.3*), where the user can decide whether to assign the **integrationID** to am existing patient, or create a new one using the specified data.

If the user decides to assign the **integrationID** to existing patients, and the data provided by the PMS differs from the patient data in 3DD, the client will show the synchronization dialog to user (*see fig.4*), which will allow the user to decide if they want to keep old data or apply new.

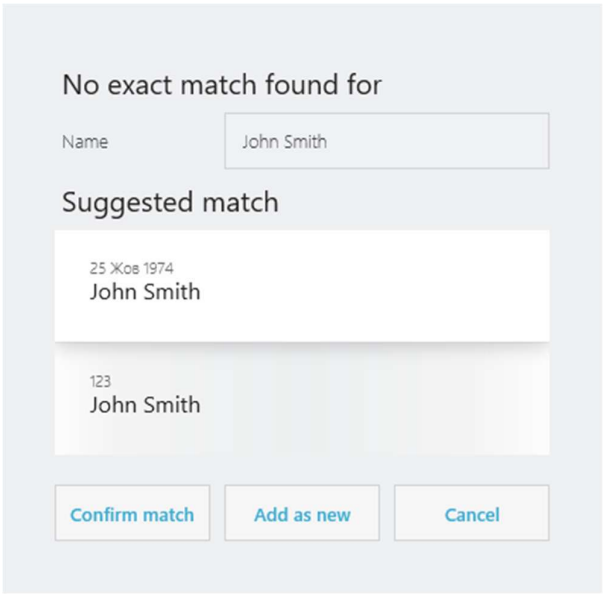

Figure 3. Patient merge dialog

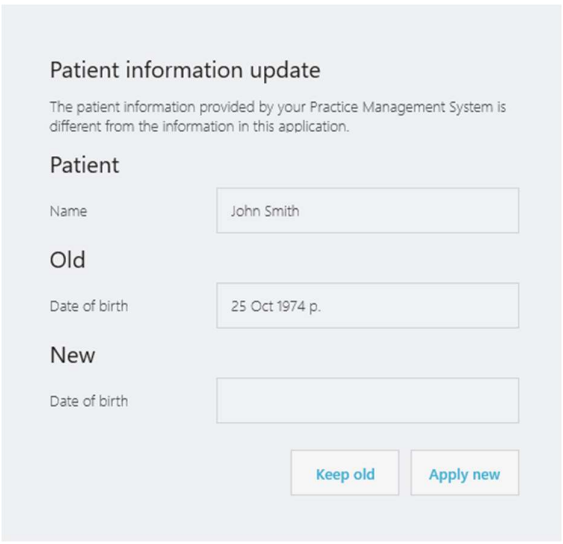

Figure 4. Patient information sync dialog

The 3DD client will automatically navigate to the page required to perform the requested operation.

Example URLs

The URL for creating a patient using full name only, on the client, with both the merging and request IDs:

```
https://localhost:5484/DentalDesktop/WebService/CreatePatient?
MergePatient=True&
FirstName=Mathew&
LastName=Fox&
RequestId=2446cb63-2f79-4f4d-9da6-c24a17755bc5&
IntegrationId=123&
ClientId=5be65d15-31a2-442b-a93b-0bd65ec8abe9
```

The URL for creating a patient using full name only, on the server:

```
https://localhost:5484/DentalDesktop/WebService/CreatePatient?
FirstName=Andrew&
LastName=Hulshult&
IntegrationId=123
```

/UpdatePatient

Summary

This method allows information to be updated for already existing 3DD patient, with or without the 3DD client involvement.

Input

The full URL to call the method with example parameters:

```
https://localhost:5484/DentalDesktop/WebService/UpdatePatient?
FirstName=John&
LastName=Smith&
DateOfBirth=14.11.2019&
PatientId=123&
IntegrationId=123&
ClientId=5be65d15-31a2-442b-a93b-0bd65ec8abe9&
RequestId=8b02f6ee-af40-458d-b1b3-b56ba604ead8
```

This method can accept the following parameters:

- **FirstName** (**String**) – updated first name value of the patient;
- **LastName*** (**String**) – updated last name value of the patient;
- **DateOfBirth** (DateTime) – updated date of birth value of the patient;
- **PatientId** (**String**) – can be used to specify any visual patient-specific ID (e.g. SSN);
- **IntegrationId*** (**String**) - ID of the patient that should be updated;
- **ClientId** (**GUID**) – ID of the 3DD client where the patient update will occur. If the ID is not specified, or set to the default empty GUID, the operation will be performed on the server;
- **RequestId** (**GUID**) - optional ID for the operation that can be used to identify the operation status;

The parameters marked by ***** asterisk are mandatory and the other parameters can be omitted (*those parameters will be considered as empty*).

Output

This method does not provide any output.

Remarks

The parameters of this method must satisfy the following conditions:

- **IntegrationId** parameter must be specified, and its value must correspond to the integration ID of the existing patient;
- **LastName** parameter must be specified and not have an empty value;
- If the **PatientId** parameter is specified, its value must be unique to the patient to be updated.
- If any of those conditions are not satisfied, the call to the method will fail, creating am corresponding request status (*see **GetRequestStatusById** method for details*).

If the **ClientId** is not specified, the patient update operation will be performed on the server. If the operation is performed on the 3DD client, the 3DD client will navigate to required page (*if it is not already active*), and then it will show the synchronization dialog (*see fig.4*), which will allow the user to decide if they want to keep old data or apply a new one.

If the parameter is not specified in the URL, it still uses a default (*empty*) value, which is used as a new value for the corresponding patient data, hence it will overwrite existing data with an empty value, effectively erasing it. E.g. omitting the first name parameter from the URL will erase the first name of patient.

Example URLs
The URL for updating the patient on the server:

```
https://localhost:5484/DentalDesktop/WebService/UpdatePatient?
FirstName=John&
LastName=Smith&
IntegrationId=123
```

/SelectPatient

Summary
This method allows selecting the 3DD patient by specified integration ID on a specific 3DD client.

Input
The full URL to call the method with example parameters:

```
https://localhost:5484/DentalDesktop/WebService/SelectPatient?
IntegrationId=123&
ClientId=5be65d15-31a2-442b-a93b-0bd65ec8abe9&
RequestId=8b02f6ee-af40-458d-b1b3-b56ba604ead8
```

This method accepts the following parameters:

- **IntegrationId*** (**String**) – the ID of the patient that will be selected;
- **ClientId*** (**GUID**) – the ID of the client where the patient should be selected;
- **RequestId** (**GUID**) - optional ID for the operation that can be used to identify the operation status;

The parameters marked by * asterisk are mandatory, the other parameters can be omitted, in which case the default value will be used.

Output
This method does not provide any output.

Remarks
The operation is supported only on the client. The **3DD must have an existing patient with a corresponding integration ID.** Patient selection can only occur on certain pages, hence the 3DD client will navigate to this page if it is not already active.

/CreateCase

Summary
This method starts a new case workflow for 3DD patients with a specific integration ID on specific 3DD client.

Input
The full URL to call the method with examples of parameters:

```
https://localhost:5484/DentalDesktop/WebService/CreateCase?
IntegrationId=123&
```

ClientId=5be65d15-31a2-442b-a93b-0bd65ec8abe9&
RequestId=8b02f6ee-af40-458d-b1b3-b56ba604ead8

This method accepts the following parameters:

- **IntegrationId*** (**String**) – ID of the patient that will be used as the owner for a new case;
- **ClientId*** (**GUID**) – ID of the client where a new case workflow should be started;
- **RequestId** (**GUID**) - optional ID for the operation that can be used to identify the operation's status;

The parameters marked by * asterisk are mandatory, the other parameters can be omitted, in which case a default value will be used.

Output
This method does not provide any output.

Remarks
The operation is supported only on the client. The **3DD must have an existing patient with a corresponding integration ID.** Once performed, the user will be navigated to a new case page, where they can setup a new case workflow (*in same manner as if the 3DD user would start a new case workflow by themselves*).

/SelectCase

Summary
This method does a case selection by its ID on specific 3DD client.

Input
The full URL to call the method with an example of the parameters:

https://localhost:5484/DentalDesktop/WebService/SelectCase?
CaseId=7d728802-d04c-4d87-bf7d-dcd48b48fd93&
ClientId=5be65d15-31a2-442b-a93b-0bd65ec8abe9&
RequestId=e75b88c2-ac03-42d5-aed0-225207f28d1e

This method accepts the following parameters:

- **CaseId*** (**GUID**) – the ID of the case to be selected;
- **ClientId*** (**GUID**) – the ID of the client where the case should be selected;
- **RequestId** (**GUID**) – the optional ID for the operation that can be used to identify the operation's status;

The parameters marked by * asterisk is mandatory, the other parameters can be omitted, in which case a default value will be used.

Output
This method does not provide any output.

Remarks

The operation is only supported on the client. 3DD must have an existing case with the corresponding case ID. If the current page supports case selection, the case is then selected on the current page, otherwise the user will be automatically taken to the homepage (*configured in user settings*).

/GetRequestStatusById

Summary

This method allows information to be retrieved about the operation's execution status.

Input

The full URL to call the method with an example of the parameters:

https://localhost:5484/DentalDesktop/WebService/GetRequestStatusById?Id=e75b88c2-ac03-42d5-aed0-225207f28d1e

This method accepts the following parameters:

- **Id** (**GUID**) – ID of the request to provide its status;

Output

The XML string that can be deserialized to the instance of RequestStatus containing the information about the current status of the operation with the specified ID.

An example of the received output:

```
▼<RequestStatus xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <RequestId>00000000-0000-0000-0000-000000000000</RequestId>
   <StatusCode>101</StatusCode>
  ▼<StatusDescription>
     You should set at least integration ID for creating case request.
   </StatusDescription>
 </RequestStatus>
```

RequestStatus class contains the following fields:

- **RequestId** (**GUID**) – the ID of the corresponding request;
- **StatusCode** (**Integer**) – the numeric representation of the result status, that can be used for the programmatic feedback;
- **StatusDescription** (**String**) – readable technical description of the request status (always in English);

Remarks

This method was designed to work along with with any methods supporting the **RequestId** parameter. **GetRequestStatusById** method allows to track the status of the operation by a value of the **RequestId** parameter, that was used when the operation was invoked. If the operation is invoked without a specified **RequestId** parameter, the request status for the operation will still be created, using the empty GUID value. Also, if the operation fails, the status information will contain exact the reason of failure.

The status is saved only for the last 10 requests. If this limit is exceeded, then the oldest entries are removed.

If two operations are invoked with same the **RequestId**, this method will return a result for the last of them.

To see some of the possible results and their descriptions, please refer to the "**Request status reference**" section.

/RemoveRequestById

Summary
This method allows request removal from the requests pool on the server before it was retrieved by the client for execution.

Input
The full URL to call the method with an example of parameters:

```
https://localhost:5484/DentalDesktop/WebService/RemoveRequestById?
Id=e75b88c2-ac03-42d5-aed0-225207f28d1e
```

This method accepts the following parameters:

- **Id** (**GUID**) – the ID of the request to be removed from the pool;

Output
A Boolean value indicating if the request was removed (*false if request with such ID was not found on server*).

Remarks
The main purpose of this method is removal of requests that are "stuck" in the pool, e.g. that have the **ClientId** property not corresponding to any connected clients.

/GetCasesForPatient

Summary
This method allows user to search for information about cases of specific patients by defined parameters.

Input
The full URL to call the method with an example of the parameters:

```
https://localhost:5484/DentalDesktop/WebService/GetCasesForPatient?
IntegrationId=123&
CreationDateFrom=2018-04-30&
CreationDateTo=2019-02-01&
DeliveryDateFrom=2020-01-01&
DelivertyDateTo=2020-06-05&
IsCreated=True&
IsScanned=False&
IsModelled=False
```

This method accepts the following parameters:

- **IntegrationId** (**String**) – the ID of the patient whose cases operation needs searching;
- **CreationDateFrom** (**DateTime**) – allows to filter out the cases by a minimal creation date value;

- **`CreationDateTo`** (`DateTime`) – allows to filter out the cases by a maximal creation date value;
- **`DeliveryDateFrom`** (`DateTime`) – allows to filter out the cases by a minimal delivery date value;
- **`DeliveryDateTo`** (`DateTime`) – allows to filter out the cases by a maximal delivery date value;
- **`IsCreated`** (`Boolean`) – if it's specified and is set to true, the operation will search for the cases with the "Created" status in Dental Desktop (*not yet scanned cases*);
- **`IsScanned`** (`Boolean`) – if it's specified and is set to true, the operation will search for the cases with "Scanned" status in Dental Desktop (*cases with scans yet to be modelled*);
- **`IsModelled`** (`Boolean`) – if it's specified and is set to true, the operation will search for cases with "Modelled" status in Dental Desktop (*fully modelled cases*);

The method allows any parameter except the **`IntegrationId`** to be omitted.

Output

The XML string that can be deserialized to collect the `CaseInfo` entries containing the information about the cases found.

An example of the method's output:


```
▼<ArrayOfCaseInfo xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  ▼<CaseInfo>
     <CaseId>7d728802-d04c-4d87-bf7d-dcd48b48fd93</CaseId>
     <PatientName>John Smith</PatientName>
     <PatientIntegrationId>123</PatientIntegrationId>
     <SentToLab>My CoDx</SentToLab>
     <CaseSentDate>2019-11-18T10:54:22+02:00</CaseSentDate>
     <DeliveryDate>2019-11-14T11:00:00Z</DeliveryDate>
    ▼<Indications>
      ▼<IndicationInfo>
         <From>3</From>
         <To>3</To>
         <IndicationName>Crown</IndicationName>
         <Shade>Wax A1</Shade>
       </IndicationInfo>
     </Indications>
    ▼<AttachedImages>
      ▼<AttachedImageInfo>
         <Name>74d3ac7f-603f-4c4f-9379-f322f0bcc0c0.jpeg</Name>
         <Id>8911a8ff-64c7-44cb-8ab1-45fdd9122397</Id>
         <TimeStamp>2019-11-14T11:18:12Z</TimeStamp>
       </AttachedImageInfo>
     </AttachedImages>
    ▼<SurfaceScans>
      ▼<SurfaceScanInfo>
         <Id>24748b06-963f-479e-92a4-d9e295680d88</Id>
         <From>17</From>
         <To>32</To>
         <ScanType>Jaw</ScanType>
         <Sectioned>false</Sectioned>
         <PhysicallyPrepared>false</PhysicallyPrepared>
         <TimeStamp>2019-11-13T09:39:15Z</TimeStamp>
       </SurfaceScanInfo>
      ▼<SurfaceScanInfo>
         <Id>efd1f89d-0e42-47bb-a09a-eac8310e3564</Id>
         <From>1</From>
         <To>16</To>
         <ScanType>Jaw</ScanType>
         <Sectioned>false</Sectioned>
         <PhysicallyPrepared>true</PhysicallyPrepared>
         <TimeStamp>2019-11-13T09:39:15Z</TimeStamp>
       </SurfaceScanInfo>
     </SurfaceScans>
   </CaseInfo>
</ArrayOfCaseInfo>
```

`CaseInfo` class contains the following fields:

- **`AttachedImages`** – a collection of `AttachedImageInfo` entries, containing information about images attached in the case (*see `AttachedImageInfo` reference below*);
- **`CaseCreationDate`** (`DateTime`) – the time when the case was created;
- **`CaseId`** (`GUID`) – ID of the case;
- **`CaseSentDate`** (`DateTime`) – the date when this case was sent to the receiver, it will return a minimal value if the case was never sent;
- **`DeliveryDate`** (`DateTime`) – the date when this item will be delivered to the patient;
- **`Indications`** – a collection of the `IndicationInfo` entries, containing information about the indications the case (*see `IndicationInfo` reference below*);
- **`IsModelled`** (`Boolean`) – the property indicating if the case was modelled;
- **`PatientIntegrationId`** (`String`) – the integration ID of the patient associated with the case;
- **`PatientName`** (`String`) – a full name of the patient associated with the case;
- **`SentToLab`** (`String`) – the name of the 3Shape Communicate receiver this case will be sent to *("In-house" if case will be not sent*);
- **`SurfaceScans`** – a collection of `SurfaceScanInfo` entries, containing information about the surface scans in the case (*see `SurfaceScanInfo` reference below*);

`AttachedImageInfo` class contains the following fields:

- **`Id`** (`GUID`) – the ID of the attached image file;
- **`Name`** (`String`) – the filename of the attached image;

`IndicationInfo` class contains the following fields:

- **`From`** (`Integer`) – the number of the first tooth for indication (*in UNN*);
- **`To`** (`Integer`) – the number of the last tooth for indication (*in UNN*);
    - If the indication occupies a single tooth, **`From`** and **`To`** values will be the same;
- **`IndicationName`** (`String`) – name of the indication;
- **`Shade`** (`String`) – the shade of the indication (*including material*);

`SurfaceScanInfo` class contains the following fields:

- **`Id`** (`GUID`) – ID of the surface scan file;
- **`From`** (`Integer`) – the number of the first tooth in the scan (*in UNN*);
- **`To`** (`Integer`) – the number of the last tooth in the scan (*in UNN*);

  - **From** and **To** values can be used, for example, to determine if the jaw scan corresponds to the lower or upper jaw;

- **ScanType** (**String**) – a description of the surface scan type. Possible values are:
  - **Unknown**;
  - **Jaw**;
  - **JawWithScanFlag**;
  - **Bite**;
  - **Waxup**;
  - **WaxRim**;
  - **Tooth**;
  - **Face**;
- **Sectioned** (**Boolean**) – indicates if the scan is sectioned;
- **PhysicallyPrepared** (**Boolean**) – indicates if the scan is physically prepared (*holds reduced teeth*);
- **TimeStamp** (**DateTime**) – the last time the scan file was modified;

Remarks

If the date ranges are inconsistent *("From" is larger than "To", resulting in impossible range*), the method will return a BadRequest error code.

Specifying any of the **IsCreated, IsScanned** or **IsModelled** parameters will make the operation look for cases with corresponding statuses, ignoring the cases with the status, if it is not specified. E.g. searching for the cases with **IsCreated** and **IsModelled** will ignore any cases with a "Scanned" status. Not specifying any status criteria will result in a search for cases with all the statuses.

/GetAttachedImage

Summary

This method allows images to be fetched (*clinical photos*) from cases on the 3DD server.

Input

The full URL to call the method with an example of the parameters:

```
https://localhost:5484/DentalDesktop/WebService/GetAttachedImage?
CaseId=7d728802-d04c-4d87-bf7d-dcd48b48fd93&
AttachedImageId=8911a8ff-64c7-44cb-8ab1-45fdd9122397
```

This method accepts the following parameters:

- **CaseId** (**GUID**) – the ID of the case that contains the attached image;

- **AttachedImageID** (**GUID**) – the ID of the image in the case to be obtained; can be obtained using **GetCasesInformation**;

Output
The XML string that can be deserialized to a raw byte array containing the image data.

Remarks
This method is designed to work in combination with the **GetCasesInformation** method – PMS obtains case IDs and their attached image IDs using **GetCasesInformation**, and then fetches the actual images using the **GetAttachedImage** method.

Workflow examples

Examples of workflows for mapping the patient

1. In PMS, user selects the patient entry, and requests mapping it with the 3DD patient.
2. PMS system queries PMS API for list of the 3DD clients, that are already connected to the server.
3. PMS composes user-friendly list of the 3DD clients, and shows it to the user;
4. User selects the 3DD client they would like to perform the mapping on;
5. PMS uses selected client ID, and patient entry data, to compose the patient creation method URL, and executes it.
6. After executing the method, PMS starts pulling the operation status.
7. On the 3DD client, a merging dialog is shown.
8. The 3DD user handles the merge, selecting the proper 3DD patient.
9. PMS registers the successfully handled operation and shows the PMS user a message about the successful merge.

Example of workflow for updating patient on server

1. In PMS, the user selects a patient entry, which was recently updated, but has the non-unique patient ID (SSN), and invokes patient information synchronization.
2. PMS composes the patient update method URL and executes it.
3. After executing the method, PMS starts pulling the operation status.
4. 3DD server detects an invalid setup and creates the corresponding request status.
5. PMS registers the issue, parses error code (*not unique patient ID*), and shows a user-friendly message with an explanation to PMS user.
6. The PMS user resolves the issue ensuring that the patient ID in PMS patient entry is unique and executes the patient update request again.

7. The patient is successfully updated on the 3DD server.
8. PMS successfully registers the handled operation and shows the PMS user message about the successful update.

Example of workflow for getting attached images from cases

1. In PMS, the user selects the patient, and requests information about the attached images of all its cases.
2. PMS composes cases search method using the patient integration ID, and searches for all the patient cases.
3. By having all patient cases, their attached images, and corresponding IDs, PMS fetches images from the 3DD using the corresponding method.
4. PMS shows all the attached images to user.

Request status reference

Code	Description	Notes
-1	The request was received by the server and awaits processing by the client.	Due to the periodic nature of the request handing, this status is assigned to every request before the client gets it from the server.
0	The request was accepted, and will be processed by the client	Assigned once the request is obtained by the client.
1	The request was rejected on the client side.	Assigned if the user cancels the process during the 3DD side operation (*e.g. closes patients merging dialog*).
2	The request was rejected on the client side, due to failure in page navigation.	Assigned if 3DD was unable to navigate to the required page.
3	The request was removed before it was handled.	Assigned if the request was removed from the server using a corresponding method.
4	The request was processed.	Assigned once the request was processed (e.g. patient was successfully created or updated).
5	The request is not supported	Assigned if the request is valid, but is not supported by receiver (e.g. if client-only request is sent to server)
6	The request cannot be handled, the client is not responding.	Assigned if the Dental Desktop client was not able to process the request for some time.
10	Could not process the request due to an exception (<exception>)	Assigned if the exception happens on the 3DD side. The description will contain the full exception text.

101	You should set at least the integration ID for creating a case request.	Assigned if the method URL is missing a parameter, that is mandatory for the called method.
102	You should set at least the last name to search for existing patients or create new.	
103	You should set at least the case ID to view the case	
104	No cases found with the specified case ID	Assigned if no case with that ID exists in the 3DD database.
105	SSN should be unique for the patient.	Assigned if there is already a patient with the specified patient ID on the server.
106	No patient was found with the specified patient ID	Assigned for the **`UpdatePatient`** request if no patient with the specified integration ID exists.
107	The patient with the same patient ID already exists	Assigned for the **`CreatePatient`** request if the patient with specified integration ID already exists.
108	Internal error	Assigned if the Dental Desktop client encountered some internal error.
404	A request with the specified ID was not found on the server.	Assigned if no request with that ID was created before or was created more than ten requests ago.

Command Line Integration

The Command Line integration (CLI) is best supported by allowing users to create custom buttons in the PMS software. If the PMS supports this functionality, the work from the PMS supplier should mostly/only be to provide data-field names to the customers. This can be done in many ways, through a quick-guide/video guide.

Important note:

- CLI currently requires both the PMS and Dental Desktop (3DD) to be installed on the same computer.

Supported workflows

- Creating and updating patients in 3DD.

Overall workflow

1. To transfer patient data from a PMS to 3DD, the PMS user, with 3DD installed, configures a command line request in the PMS (*one-time operation*),
2. Using the configured command line request, the PMS user sends the patient data to 3DD,
3. Once the command is executed, the patient data is transferred and becomes available in 3DD, this also opens 3DD with the patient selected,
4. The PMS user can also update the patient data in 3DD from the PMS software using the same command.

Setting up integration with Command Line (e.g., Dolphin)

For the integration with Command Line, the `DentalDesktopCmd.exe` executable is used. By default, it is located in: `C:\Program Files\3Shape\Dental Desktop\Plugins\ThreeShape.PMSIntegration\DentalDesktopCmd.exe`.

If the default Dental Desktop installation path is changed (or `C:\` drive is not present), it can be found using this path template `{PathToDentalDesktopInstallation}\Dental Desktop\Plugins\ThreeShape.PMSIntegration\DentalDesktopCmd.exe`.

List of available arguments

Argument	Value
`-integrationid`	PMS software patient number/id which is used to uniquely identify the patient in the partner software (e.g. `-integrationid=123456789`).
`-firstname`	First name of the patient (e.g. `-firstname=John`).
`-lastname`	Last name of the patient (e.g. `-firstname=Doe`).

-birthday	Date of birth of the patient in the format yyyymmdd or using the current short date format of Windows (e.g. -birthday=19820126).
-patientid	A specific patientid, could be a social security number (SSN) or a similar number/id of patient, and is usually given by the external agency (e.g. -patientid=12345).

Example of Command Line request

```
"C:\Program Files\3Shape\Dental Desktop\Plugins\ThreeShape.PMSIntegration\DentalDesktopCmd.exe" -integrationid="IntegrationId"-firstName="FirstName" -lastName="LastName" -patientid="PatientId"
```

Important note:

- Patient ID and last name are mandatory information for all patients in order to be able to perform a command line request to Dental Desktop

Enabling the Command Line Integration in Dental Desktop

1. Go to More > Settings > Practice Management Integration > Setup and click the check box to enable the integration.
2. Select "General command line" as the integration type. If a confirmation dialogue appears, click 'Yes' to allow DentalDesktopCmd.exe to make changes to your device.
3. All the information necessary for completing the integration in the PMS is now provided in the "Info" section on the same page. Copy this information to the relevant place in the PMS software.
4. Finally, save the changes for the integration to take effect.

Important note:

- It is recommended to create a separate user in Dental Desktop for the integration purposes. After the user has been created, close and then reopen Dental Desktop to be able to use this newly created user.

VDDS integration

Dental Desktop (3DD) supports VDDS media version 1.4. If the PMS supports VDDS, the work from the PMS supplier should mostly/only be to inform their users, that 3DD is now supported. 3Shape will make sure to do the same once informed of the enabled connection.

Important note:

- VDDS currently requires both the PMS and 3DD to be installed on the same computer.

The following workflows are supported:

1. Transfer of patient information to Dental Desktop from a VDDS partner software system
2. Transfer of descriptions of the patient's attached images to the VDDS system
3. Transfer of images to the VDDS system.

Enabling the VDDS integration in Dental Desktop

1. Go to More > Settings > Practice Management Integration > Setup and click the check box to enable the integration.
2. Select "VDDS" as the integration type. If a confirmation dialogue appears, click **Yes** to allow DentalDesktopCmd.exe to make changes to your device.
3. Done. By enabling Practice Management Integration in Dental Desktop, Dental Desktop becomes registered inside the VDDS media and can be used for interfacing with a VDDS partner.

Important notes:

- If Dental Desktop is already opened on the PMS PC, this instance will be reused for the integration. In case of several Dental Desktop instances being opened, please close all but one.
- If Dental Desktop is not opened on the PMS PC, a new instance will be opened automatically, requiring the user to log in for the processing to be completed (the user has 5 minutes timeout to perform logging in).

About 3Shape

3Shape is changing dentistry together with dental professionals across the world by developing innovations that provide superior dental care for patients. Our portfolio of 3D scanners and CAD/CAM software solutions for the dental industry includes the multiple award-winning 3Shape TRIOS® intraoral scanner, the 3Shape X1® CBCT scanner, as well as market-leading scanning and design software solutions for both dental practices and labs.

Two graduate students founded 3Shape in Denmark's capital in the year 2000. Today, 3Shape employees serve customers in over 100 countries from 3Shape offices around the world. 3Shape's products and innovations continue to challenge traditional methods, enabling dental professionals to treat more patients more effectively. www.3shape.com

Company name
Address
Telephone
www.3shape.com
CVR Number: xx xx xx xx